18 multiplier controls

20 a first multiplier-defined restricted set A

In the section DESCRIPTION – CLAIM 1 AND THE PREFERRED EMBODIMENT beginning on page 17, the first two paragraphs of page 18 have been amended as follows:

The preferred embodiment of the invention is described in claim **1** and shown in Figure 1. It is a machine used in computing one or more sums of products, comprising a first number in a first finite-precision numeric format, a second number in a second finite-precision numeric format, and first multiplier means for computing the product of the first number and the second number.

According to Figure 1, the preferred embodiment of the invention is a non-general and non-constant multiplier **10** with a first multiplier input X **12**, a first multiplicand input Y **14**, a first multiplier output Z **16**, and multiplier controls **18**. As indicated in the block of non-general and non-constant multiplier **10**, first multiplier output Z **16** is equal to the product of first multiplier input X **12** and first multiplicand input Y **14** if and only if first multiplier input X **12** is a member of a first multiplier-defined restricted set A **20**. First multiplier-defined restricted set A **20** has cardinality of two or more but is a strict subset of all the supported representations at first multiplier input X **12**.

The first number (first multiplier input X **12** in Figure 1) has a first finite number of representation elements and the second number (first multiplicand input Y **14** in Figure 1) has a second finite number of representation elements. One type of representation element is a binary representation element which can take on two possible values. Such a representation element is often called a "bit". Other types of representation elements are possible. The invention as described by

claim **1** does not require that the first finite-precision numeric format be the same as the second finite-precision numeric format. The invention as described by claim **1** also does not require that the representation elements be binary representation elements.

The last paragraph on page 19 running over onto page 20 has been amended as follows:

The first multiplier-defined restricted set includes all the numbers having the first finite-precision numeric format for which the first multiplier means can produce correct products when multiplying by the second number. Since the first multiplier-defined restricted set does not include all numbers having the first finite-precision numeric format, the first multiplier means is not a general multiplier for numbers in the first finite-precision numeric format. Since the first multiplier-defined restricted set has more than one member, the first multiplier means is not be a constant multiplier. Membership of multiplier-defined restricted sets in the present work, including the first multiplier-defined restricted set, is determined by the multiplier processing structure, not by the values of control signals. Membership of a number representation in a multiplier-defined restricted set implies that there is at least one set of control signal values for which correct products are computed, while lack of membership implies that there are no sets of control signal values for which correct products are computed.

The first full paragraph on page 20 has been amended as follow:

A further restriction imposed by claim **1** is that at least one of the one or more sums of products in which the machine of claim **1** is used is not a desired product of two numbers. Any product of two numbers can be computed as a sum of products by separating one or both of the numbers

into equivalent sums and applying the distributive property of multiplication. The language in claim **1** restricting the sum of products being computed to not being a desired product of two numbers is intended to avoid conflict with this interpretation of product computation. Claim **1** should not cover a signal processing transform that is in fact a multiplier. Claim **1** should cover signal processing transforms that use sums of products and that are not themselves multipliers. To this end, it is required that the multiplier inputs be a transform stage weight and a transform input, with at least two different inputs appearing in the weighted sum to which the resulting product contributes.

Figures:

Applicant has furnished the drawing Figure 1 which appears separately on the next page of this Amendment.